**Public transport management**

The public transport segment uses innovative transport technologies such as passenger information systems, integrated ticketing, automated toll collection, passenger information display systems, and advanced transport logistics solutions. They all use IoT technologies that offer the following benefits:

* **Real-time vehicle tracking** for arrival time alerts via mobile devices and information displays for passengers at stops and stations.
* **Personalized transport timetable tracking** with information on fundamental changes such as delays, station closings, or re-routing.
* **Real-time data tracking**helps build the correct route due to unforeseen incidents such as accidents, road works, emergencies, etc.

**1. Hardware Setup:**

   - Acquire IoT sensors like GPS modules and passenger counters for each public transportation vehicle.

   - Install and configure these sensors in the vehicles.

**2. Connectivity:**

   - Ensure the sensors have a reliable internet connection, possibly through cellular networks or Wi-Fi.

**3. Data Collection**:

   - Develop firmware for the IoT sensors (likely using C/C++ or a platform-specific language) to gather data from GPS and passenger counters.

**4. Data Transmission**:

   - Create Python scripts on the IoT sensors to process and transmit real-time data to the transit information platform. You may need to choose a protocol for data transmission, like MQTT or HTTP.

**5. Transit Information Platform**:

   - Set up a central server or cloud-based platform to receive and process data from the sensors.

   - Implement data storage and real-time data processing capabilities.

* **Vehicle Sensors**

In ITS, identifying the type of sensors to develop applications that contribute to address problems such as: (1) traffic congestion and parking difficulties, (2) longer commuting times, (3) higher levels of CO2 emissions, and (4) increase in the number of road accidents, among others is of critical importance for improving a vehicle’s performance as well enhancing the driving experience.

### In Road Sensors

Strategic investment in transportation infrastructures is vital for a country’s growth and is the central core of a modern economy. Each year, governments worldwide spend a huge amount of money in the transportation sector. In the United States, the yearly investment is around 1.6 percent of the Gross Domestic Product (GDP) [[22](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5948625/#B22-sensors-18-01212)] and Europe invested around 102 billion euros in 2014 with 52% spent in road infrastructures [[23](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5948625/#B23-sensors-18-01212)]. Even though the automotive industry has invested a lot of money to increase safety, performance and comfort in vehicles using sensors within the vehicle; traffic data collection using mechanisms located along the roadside has become one of the main challenges for intelligent transportation systems.

### Discussion about Key Sensors

Although sensors are available and widespread, they are a small portion of the various types of equipment used in the automotive industry’s future planning: the self-driving vehicle. The redundancy and integration of sensors will improve the safety and performance of self-driving, automated or autonomous vehicles. Today’s vehicles are already equipped with radar and camera systems, redundant sensors, and software to control them.

**1. Hardware Setup:**

• You'll need lot sensors that can capture location data (e.g., GPS) and ridership data (e.g., passenger counters).

2**.Transit Information Platform:**

• Ensure you have access to a transit information platform where

the data will be sent. This platform should provide APIs or

mechanisms to receive and process the incoming data.

**3. Python Libraries:**

• Use appropriate Python libraries to interact with the hardware. For

example, if you're using a Raspberry Pi with GPS, you can use libraries like gpsd or gps3 for GPS data.

**4. Data Collection**:

Write Python code to collect real-time location data and ridership

data from the sensors. This can involve reading from GPS modules, passenger counters, or other sensors.

**5. Data Formatting:**

 Organize the data in a format suitable for transmission. It might be in JSON or another structured format.

**6. Data Transmission:**

• Use HTTP requests or MQTT (Message Queuing Telemetry

Transport) to send the data to the transit information platform. Python libraries like requests for HTTP or paho-mqtt for MQTT

can be useful.

**7. Authentication and Security:**

If the transit platform requires authentication or secure transmission, implement the necessary security measures, such

as API keys or encryption.

**8. Error Handling:**

• Implement error-handling mechanisms to deal with network

issues or platform unavailability.

**Program:**

import requests

import json

location\_data = {"latitude": 123.456, "longitude": 789.012) ridership\_data = {"passengers": 42}

transit\_platform\_url ="<https://transit-api.example.com/data-endpoint>"

data = {"location": location\_data, "ridership": ridership\_data}

response = [requests.post](http://requests.post/)(transit\_platform\_url, json-dataif response.status\_code == 200: print("Data sent successfully.")

else: print("Failed to send data. Status code: (response.status\_code}")

except Exception as e: print("Error: {e}")

|  |  |
| --- | --- |
|  |  |